INVESTIGATION OF METHODS TO OPTIMIZE CAPTURED AIR BUBBLE SURFACE EFFECT SHIP DIGITAL SIMULATION FOR IRREGULAR SEA CONDITIONS

William Ray Mitchell

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

INVESTIGATION OF METHODS TO OPTIMIZE
CAPTURED AIR BUBBLE SURFACE EFFECT SHIP
DIGITAL SIMULATION FOR IRREGULAR SEA CONDITIONS

by

William Ray Mitchell

September 1974

Thesis Advisor: A. Gerba Jr.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Investigation of Methods to Optimize Captured Air Bubble Surface Effect Ship Digital Simulation for Irregular Sea Conditions		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; September 1974 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) William Ray Mitchell		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		12. REPORT DATE September 1974 13. NUMBER OF PAGES 53
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, CA 93940		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Surface Effect Ship Captured Air Bubble Craft Computer Simulation CPU Time Savings		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A study was conducted of the digital computer simulation of the equations of motion of the Surface Effect Ship with six degrees of freedom as developed by the Oceanics Corporation. The version of the program under consideration was a simulation of the Bell Aerospace Systems 100-B Captured Air Bubble Craft (CAB) as adapted to the IBM-360/67 digital computer and operating in irregular seas. The objective of the study was to		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601 |

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(20. ABSTRACT continued)

optimize the simulation by reducing computation time required to obtain solutions to the forces and moments acting on the craft while maintaining a reasonable degree of accuracy of the output variables. CPU time savings achieved by use of the FORTRAN H compiler are documented. The dependence of CPU time and output accuracy on the tolerance levels established for the integration algorithm is discussed. CPU time savings versus output accuracy as the tolerance levels are increased is presented.

Investigation of Methods to Optimize
Captured Air Bubble Surface Effect Ship
Digital Simulation for Irregular Sea Conditions

by

William Ray Mitchell
Lieutenant Commander, United States Navy
B.S.E.E., University of Colorado, 1965

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1974

ABSTRACT

A study was conducted of the digital computer simulation of the equations of motion of the Surface Effect Ship with six degrees of freedom as developed by the Oceanics Corporation. The version of the program under consideration was a simulation of the Bell Aerospace Systems 100-B Captured Air Bubble Craft (CAB) as adapted to the IBM-360/67 digital computer and operating in irregular seas. The objective of the study was to optimize the simulation by reducing computation time required to obtain solutions to the forces and moments acting on the craft while maintaining a reasonable degree of accuracy of the output variables. CPU time savings achieved by use of the FORTRAN H compiler are documented. The dependence of CPU time and output accuracy on the tolerance levels established for the integration algorithm is discussed. CPU time savings versus output accuracy as the tolerance levels are increased is presented.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF DRAWINGS

TABLE OF SYMBOLS AND ABBREVIATIONS

INTGRL = Integration Subroutine

RHS = Subroutine for calculating the right hand side of the differential equations which describe the SES

F = Force (Subscripted to indicate direction)

g = Gravitational constant

I_{xx} = Mass moment of inertia about x-axis

I_{xz} = Mass product of inertia in x-z plane

I_{yy} = Mass moment of inertia about y-axis

I_{zz} = Mass moment of inertia about z-axis

K = Roll moment

m = Mass

m_b = Mass of air bubble

P = roll angular velocity

P_b = pressure in the bubble

Q = Pitch angular velocity

Q_{in} = Volumetric flow rate into plenum

Q_{out} = Flow rate due to leakage

R = Yaw angular velocity

U = longitudinal velocity

V = lateral velocity

W = vertical velocity

X = horizontal distance in direction of forward motion

Y = horizontal distance positive to starboard

Z = vertical distance positive downward

θ	=	pitch angle, positive bow up
ρ_a	=	standard atmospheric density
ϕ	=	roll angle, positive port side up
ψ	=	yaw angle, positive turn to starboard
$(\dot{\ })$	=	$\frac{d(\ )}{dt}$

ACKNOWLEDGEMENT

The author wishes to express his appreciation for the excellent assistance provided by Mr. William D. Ehrman of the Naval Postgraduate School's W. R. Church Computer Center. His expertise in computer programming techniques was a valuable aid to this study.

The guidance, assistance and unfailing patience provided by Professor Alex Gerba during the development of this thesis is most sincerely appreciated.

The continuous support and encouragement given by my wife plus her many hours of typing made it all possible.

I. INTRODUCTION

A. BACKGROUND

In recent years the U. S. Navy has developed an interest in the Surface Effect Ship (SES) concept for combatant and support type craft. As a result of this interest, Oceanics Incorporated was commissioned by the U. S. Navy to develop a digital computer simulation of the six degrees of freedom equations for SES craft that would yield time domain outputs of loads and motions. This initial report was delivered to the Navy Surface Effects Ships Project Office (SESPO) and dated August, 1971 [Ref. 1].

The simulation program was designed to achieve a sufficiently accurate and stable solution to the loads and motions equations for the modeling of the craft. The desire for real-time solutions was acknowledged, as they would be valuable for manned control simulation studies and greatly reduce costly computer time during case studies [Ref. 1].

The program was developed using FORTRAN IV computer language and was prepared in modular form for ease in adaptation to various SES configurations. The program delivered to the Naval Postgraduate School in October, 1972, was a simulation of the Bell Aerospace Systems 100-B Captured Air Bubble Craft (CAB), a 100 ton craft.

Results of simulated runs conducted by Oceanics, utilizing a Control Data Corporation 6600 digital computer

(using the scope 3.3 operating system) indicated that problem time to computer time ratios, as low as 3:1 for calm water runs and as large as 1:4 for large amplitude, oblique, irregular sea cases at high speed, could be expected [Ref. 2].

The initial studies at the Naval Postgraduate School, using the IBM-360/67 computer with the standard Fortran G compiler to solve the 100-B loads and motion program for sea state conditions, resulted in problem time to computer time ratios of 1:40 to 1:70, depending on the type run simulated. As the sea state condition increased, the computer time increased to the point where for certain type runs, 2 to 3 hours of computer time were needed for problem solution [Ref. 3]. For example, broaching studies simulating 65 knots with sea state 3 (irregular waves) at 150° encounter angle and a 30 second run time, required 121 minutes of CPU time. It should be noted however, that in the broaching study, the output requirements included a listing of sidewall gap each time subroutine RHS is entered thus adding a significant amount of CPU time.

B. OBJECTIVES

The high ratio of problem time to computer time for sea state operations demanded that an in depth study be made of the Oceanics SES simulation program as applied to the IBM-360/67 located at the Naval Postgraduate School's W.R. Church Computer Facility with the following objectives in mind:

1.) Careful analysis of each subroutine to determine those areas where the bulk of the computer time is being spent.

2.) Determine methods whereby the computational efficiency of the program, can be maximized with minimum loss in accuracy.

This thesis examines the SES 100-B digital computer simulation program on the IBM-360/67 computer with the above mentioned objectives in mind and makes recommendations for substantially reducing computer time while maintaining a reasonable degree of accuracy with regard to the loads and motion simulation.

II. INTEGRATION METHOD

A. INTRODUCTION

The SES simulation program is based on a mathematical description of the craft in six degrees of freedom as reported by Kaplan, Bentson, and Sargent [Ref. 1]. The resulting equations, in a form suitable for numerical integration on a digital computer, are given by

$$\dot{U} = \frac{F_x}{m} \tag{1}$$

$$\dot{V} = \frac{F_y}{m} - RU \tag{2}$$

$$\dot{W} = \frac{F_z}{m} \tag{3}$$

$$\dot{P} = \frac{F_k \cdot I_{zz}}{I_{xx}I_{zz} - I_{xz}^2} + \frac{F_n \cdot I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} \tag{4}$$

$$\dot{Q} = \frac{F_m}{I_{yy}} \tag{5}$$

$$\dot{R} = \frac{F_N I_{xx}}{I_{xx}I_{zz} - I_{xz}^2} + \frac{F_k I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} \tag{6}$$

$$\dot{\phi} = P \tag{7}$$

$$\dot{\theta} = Q \tag{8}$$

$$\dot{z} = w \tag{9}$$

$$\dot{x} = u \tag{10}$$

$$\dot{y} = v \tag{11}$$

$$\dot{\psi} = R \tag{12}$$

$$\dot{M}_b = \rho_a(Q_{in} - Q_{out}) \tag{13}$$

where X, Y, and Z, are the 3-dimensional spatial coordinates; P, Q, and R, are the angular velocities about the X, Y, and Z, axis respectively; I is the mass moment of inertia, and F is the force. M_b is the mass of bubble air in the plenum; Q_{in} is the volumetric quantity of air flow rate into the bubble; Q_{out} is the flow rate due to leakage, and ρ_a is the standard atmospheric reference density.

The program is of modular design utilizing a partitioning technique on the craft to calculate the various forces and moments exerted on the hull as it moves through the water. The forces and moments acting on the craft are computed in various subroutines using initial conditions supplied by subroutine INCON. Subroutine RHS consists of a simple summing algorithm to compute the total of the forces and moments in six degrees of freedom.

Subroutine INTGRL calls subroutine RHS for the calculation of the right hand side of equations defined by the Runge-Kutta-Merson (RKM) algorithm for each required step size. The elements of each k vector in the RKM algorithm are the ten variables defined by equations (1) through (9) and equation (13).

Since the variables X, Y, and ψ defined in equations (10), (11), and (12), are expected to have slow variation, it was decided that a simple means of integration using Simpson's Rule could be applied to these equations. These variables are calculated in the main program [Refs. 1 and 2].

A complete description of the computer program including its organization, flow charts, input and output description, program listing, and a user's manual for the operation of the program are available in Refs. 2 and 4.

B. SUBROUTINE INTGRL

The numerical integration technique used in subroutine INTGRL is the Runge-Kutta-Merson (RKM) method for solution of a system of first order differential equations of the form

$$\dot{y} = f(t,y) \tag{14}$$

the recursion formula is

$$Y_{n+1} = Y_n + \frac{k_1 + 4k_4 + k_5}{2} + o(h^5) \quad . \tag{15}$$

where

$$k_1 = \frac{h}{3} f(t_n, Y_n) \tag{16}$$

$$k_2 = \frac{h}{3} f(t_n + \frac{h}{3} , Y_n + k_1) \tag{17}$$

$$k_3 = \frac{h}{3} f(t_n + \frac{h}{3} , Y_n + \frac{k_1}{2} + \frac{k_2}{2}) \tag{18}$$

$$k_4 = \frac{h}{3} f(t_n + h , Y_n + \frac{k_1}{8} + \frac{9k_3}{8}) \tag{19}$$

$$k_5 = \frac{h}{3} f(t_n + h , Y_n + \frac{3k_1}{2} - \frac{9k_3}{2} + 6k_4) \tag{20}$$

and h is the stepsize. The truncation error in (15) is given by

$$\varepsilon = k_1 - \frac{9k_3}{2} + 4k_4 - \frac{k_5}{2} \tag{21}$$

Initial conditions are read in through subroutine INCON to establish the starting point for each of the ten variables plus a trial stepsize (h) and a maximum error tolerance level for each integrator. Computations are then carried out and the truncation error given by equation (21) is computed. If any element in the error vector ε is greater than the corresponding maximum tolerance level then the trial time stepwise is halved and the computations are repeated. If the error is less than the tolerance level, but greater than 1/16 this level, the computations proceed in accordance with the RKM algorithm. If all elements in ε are less than or equal to 1/16 the maximum tolerance level, the stepsize is doubled in the next integration step. This procedure automatically adjusts the time step in the computation to reflect the nature of the perturbations being experienced by the craft.

As a result, for slowly-varying phenomena (e.g., calm water, slow speed simulation), the computations are carried out quickly since larger time steps are taken, while high frequency effects (e.g., heavy seas, high speed simulations), will cause an increase in computation time.

III. PROCEDURE

A. INTRODUCTION

The obvious starting point for an efficiency study of the SES simulation program was to gain a thorough knowledge of the force and moment algorithm used. The object was to study the interconnections and interactions between the various subroutines with a goal of arriving at a more efficient (i.e. less time consuming) solution by reprogramming and/or better utilization of the IBM-360 software.

A major reprogramming effort was considered to be beyond the scope of this thesis. However, investigation of subroutines which were known to consume the bulk of the computation time was conducted. In conjunction with this investigation various timer functions built into the IBM-360 were utilized to pin point time consuming procedures.

B. IBM-360 COMPILER

The SES simulation program was originally adapted to the Naval Postgraduate School IBM-360 computer utilizing the standard FORTRAN G compiler. This compiler features a DEBUG facility and core optimizations which lends itself to the smaller job, teaching environment prevalent at the school [Ref. 5]. Among other compilers, the IBM-360 computer system has available FORTRAN H, an IBM processor which produces a highly optimized code which is highly desirable for large

production programs such as the SES simulation program [Ref. 6].

Compilation of the program in FORTRAN H resulted in an average CPU time savings of thirty-seven percent over the FORTRAN G compiler while yielding exactly the same output variable values. For example, a simulation of a thirty second run in a state three, head sea compiled in FORTRAN G required 80.5 minutes of CPU time. The same simulated run time compiled in FORTRAN H required only 50.7 minutes of CPU time. A number of runs, under varying initial conditions, were made to verify the CPU time savings using the FORTRAN H compiler. Unless otherwise indicated all further runs discussed in this thesis were made using the FORTRAN H compiler.

C. TIMER SUBROUTINES

The IBM-360/67 at the Naval Postgraduate School's W.R. Church Computer Facility has several built in timer subroutines which were helpful in determining those areas in the program in which the bulk of the CPU time was being used. Two such routines were used in this study.

1. Subroutine IONUM

Subroutine IONUM is a program which monitors the number of times an input or output device is accessed during the run of a computer program. In adapting the Oceanics simulation program to the IBM-360, extensive use of discs as temporary storage devices was made. By monitoring the

number of input/output operations with IONUM a more efficient data blocksize for the discs could be determined, thus reducing the number of accesses required.

2. Subroutine PROGLOOK

Several areas of the SES simulation program were suspected to be inefficient and/or requiring excessive time during the program solution. In order to pin point these areas of the program, subroutine PROGLOOK was used.

PROGLOOK, when linked to the program under consideration, monitors the sequential flow of the program instructions. The routine "strobes" the instruction flow at a rate of 50 samples per second or approximately once every 4000 instructions. In this manner a time history of the program functions was obtained. The output of subroutine PROGLOOK, which was of interest to this study, is in the form of a histogram which gives a plot of percentage of time the program spent performing various functions. Correlation of the location of these functions with a computer map listing of the SES simulation program served to point out time consuming areas within the program.

D. INTEGRATOR TOLERANCES

A prime suspect area of the simulation program for excess time consumption was the Runge-Kutta-Merson (RKM) algorithm used to solve 10 of the force and moment differential equations. If during a given time increment, any one of the 10 error function values is found to be outside the given

tolerance, the stepsize is halved and all variables are computed using the new stepsize. This process is continued until all 10 error functions are within their respective tolerances and then a new time increment is started.

The above stated procedure does not constitute an inordinate amount of computer time under calm water simulations as the variable values tend to vary slowly and the RKM algorithm converges rapidly to the established tolerance values. However, craft perturbations caused by such conditions as simulations of rapid turns, irregular waves to simulate sea state, and loss of a propulsion unit, can cause some of the variables to vary at a high rate of change, thus requiring many iterations of the algorithm for convergence to the established tolerance levels and resulting in large CPU times.

Reference 3 established the standard maximum tolerance levels for each integrator according to procedures set forth in Ref. 2. These tolerance levels were considered to be the basis of accuracy desired for the SES simulation program used at the Naval Postgraduate School and are listed in Table I. The input and output variables for each of the 13 integrators are shown in Fig. 1.

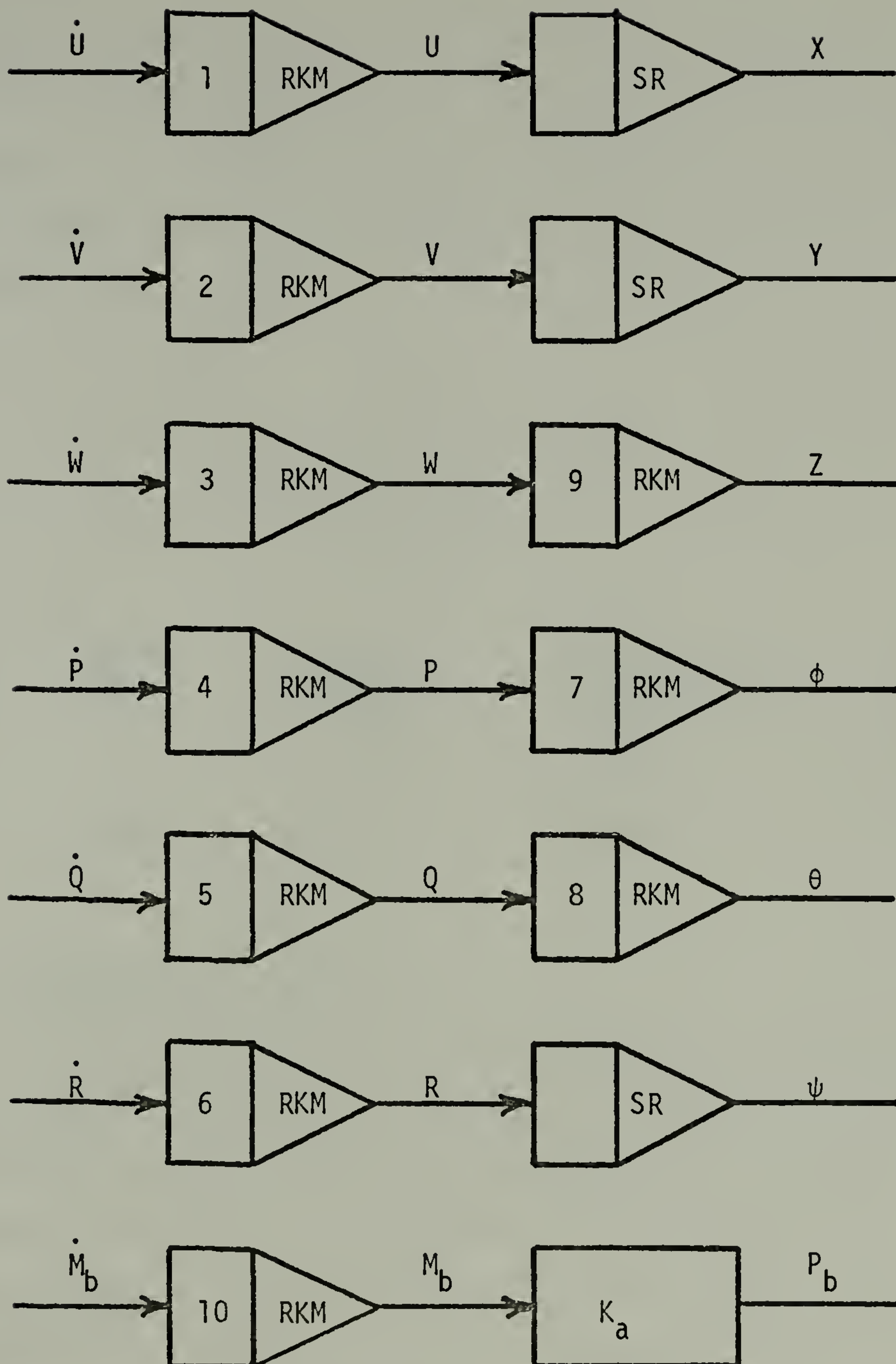
Table I

Tolerance Values For Integrators

Integrator Number	Output Variables	Tolerance Values
1	U	.0001
2	V	.0002
3	W	.000001
4	P	.0000001
5	Q	.0001
6	R	.000001
7	ϕ	.000000001
8	θ	.000000001
9	Z	.0001
10	M_b	.0001

1. Steady State Conditions

A number of runs were conducted initially, for each simulation study made, in order to establish steady state conditions for a given set of initial conditions. Reasonably accurate steady state conditions are essential to prevent initial transients in the simulation program which can cause the program to abort, or worse yet, to give erroneous results. The key variables to initialize for a given input are thrust, draft, plenum pressure, and to a lesser degree pitch angle.



RKM Runge-Kutta-Merson Method

SR Simpson's Rule

Figure 1. Inputs and Outputs of Integrators

Two run conditions were selected to fulfill the objectives of this thesis. (1) Sea state 3, 40 knot speed, head seas, and (2) Sea state 3, 60 knot speed with turn. Table II gives the initial conditions for these two run conditions.

Table II

Initial Conditions

(Sea State 3)

Speed (kts)	Draft (In)	Plenum Pressure (PSF)	Thrust (one Engine) (Lbs)	Pitch Angle (Deg)
40.0	30.0	78.0	7000.0	0.50
60.0	28.0	80.0	10800.0	0.45

2. Integrators 3 and 10

The greatest effect of the sea state condition on the SES program computation occurs on integrator 3 (heave accelerations) and integrator 10 (time rate of change of the bubble air mass). The majority of the forces and moments acting on the craft when encountering sea states is manifested in changes in Z directed accelerations and plenum volume (i.e., bubble mass).

As a first approach to reducing the CPU time for sea state simulations, the changing of the implementation of the RKM algorithm was considered. It was felt that if

Integrator 3 and 10 could be isolated and their solutions made to converge before the other eight integrations were attempted, then a considerable computation time savings would be gained due to the reduced number of iterations needed through subroutines INTGRL and RHS. However, due to the functional dependence of heave acceleration and bubble mass on some of the variables, it was found that Integrators 3 and 10 could not be made to converge independently of the other eight.

Since both Integrator 3 and 10 were extremely sensitive to craft perturbations the decision was made to loosen the tolerances on these two integrators and compare the results with those obtained using the standard tolerances (See Table I). It was felt that if the percentage change in variable values was not too great (less than 10 percent), acceptable results could be obtained with a considerable CPU time savings.

The two integrator tolerance levels were increased, in tandom, by a factor of ten in steps of one for a total of ten runs under a given set of initial conditions. Each computer run simulated a 30 second craft run and parameter values were printed out every 0.05 seconds for a sample rate of 20 per second per variable. In addition, the root mean square (RMS) value and the average value of the parameters was calculated over the entire run interval for

comparison with values obtained from runs using the standard tolerances. Using these results an error analysis could be conducted.

3. <u>Integrators 1,2 and 4-9</u>

Analysis of the CPU time versus output accuracy as observed in the above runs indicated an optimum tolerance of 0.000007 for integrator 3 and 0.0007 for integrator 10. It was then decided to investigate the sensitivity of the other eight integrators. The method used for this investigation was to hold the above tolerances on integrators 3 and 10 while increasing one of the other integrator tolerances by a factor of ten (the other seven tolerances remaining at standard values). Each integrator was tested in succession and the results compared to standard runs.

IV. DISCUSSION AND EVALUATION OF RESULTS

A. FORTRAN H COMPILER

The efficiency of the FORTRAN H compiler can best be illustrated by the following example. Reference 3 cited the disparity in CPU times between simulation runs using control statement 00201 and 00202. These runs were made using the FORTRAN G compiler. Control statement 00201 inputs directly to subroutine INCON the overall weight of the craft plus the location of the center of gravity and the mass moment of inertia about the X, Y, Z, and XZ-axis. Control statement 00202 inputs distributed weights and moments and subroutine INCON must then compute the craft weight, CG and moments. It would appear that the additional computations involved in using control statement 00202 would require more CPU time than statement 00201.

Results of Ref. 3 indicated that the opposite result was obtained and that CPU time decreased by approximately 10 percent when control statement 00202 was used as input. The reason for this result has not been established and should be looked into by future investigators. Nonetheless, results of runs compiled in FORTRAN H reduced the CPU time difference cited in Ref. 3 to less than 1 percent which illustrates the optimization capabilities of this compiler.

B. INTEGRATOR TOLERANCES

The initial runs were made to verify the sensitivity of the Runge-Kutta-Merson algorithm to integrator tolerances. The runs simulated 40 knots speed, state 3 seas, head waves, 30 second run, and confirmed that integrators 3 and 10 were the most sensitive to craft perturbations. The results of these runs were as follows.

Table III

CPU Time Comparison

(Sea State 3)

Integrator Tolerance No. 3	Integrator Tolerance No. 10	CPU Time (Minutes)	PERCENTAGE Decrease In CPU Time
0.000001*	0.0001*	53.55	Standard
0.000002	0.0002	38.74	27.66
0.000003	0.0003	33.39	37.65
0.000004	0.0004	31.38	41.40
0.000005	0.0005	29.20	45.47
0.000006	0.0006	28.90	45.73
0.000007	0.0007	26.99	49.56
0.000008	0.0008	26.19	51.09
0.000009	0.0009	25.03	53.26
0.000010	0.0010	24.50	54.25

* Standard for comparison

1. Integrator 3 and 10

The tolerances on integrators 3 and 10 were loosened during the above stated runs while the other eight integrator tolerances were maintained at their standard values. Table III shows the results of these runs with regard to CPU time and Figure 2 is a graphical representation of the data.

As the integrator tolerances were varied the RMS and average values of the output variables were compared with those obtained using standard integrator tolerances. The percentage deviations from standard values were computed in order to have a measure of output variable accuracy as a function of CPU time savings. Table IV tabulates these results for maximum tolerance error levels.

As was expected, the highest percentage deviations occured with heave acceleration (integrator 3) and the variables concerned with air flow (integrator 10). All other variables had insignificant deviations (less than 1%) from standard values. In general, the percentage deviation of the output variables increased almost linearly as the integrator tolerances were loosened, with the maximum deviation occurring for maximum tolerance level.

Increasing the Integrator tolerance levels above a factor of seven did not greatly decrease the CPU time required for problem solution but did tend to decrease the accuracy of the affected output variables. For example, the initial

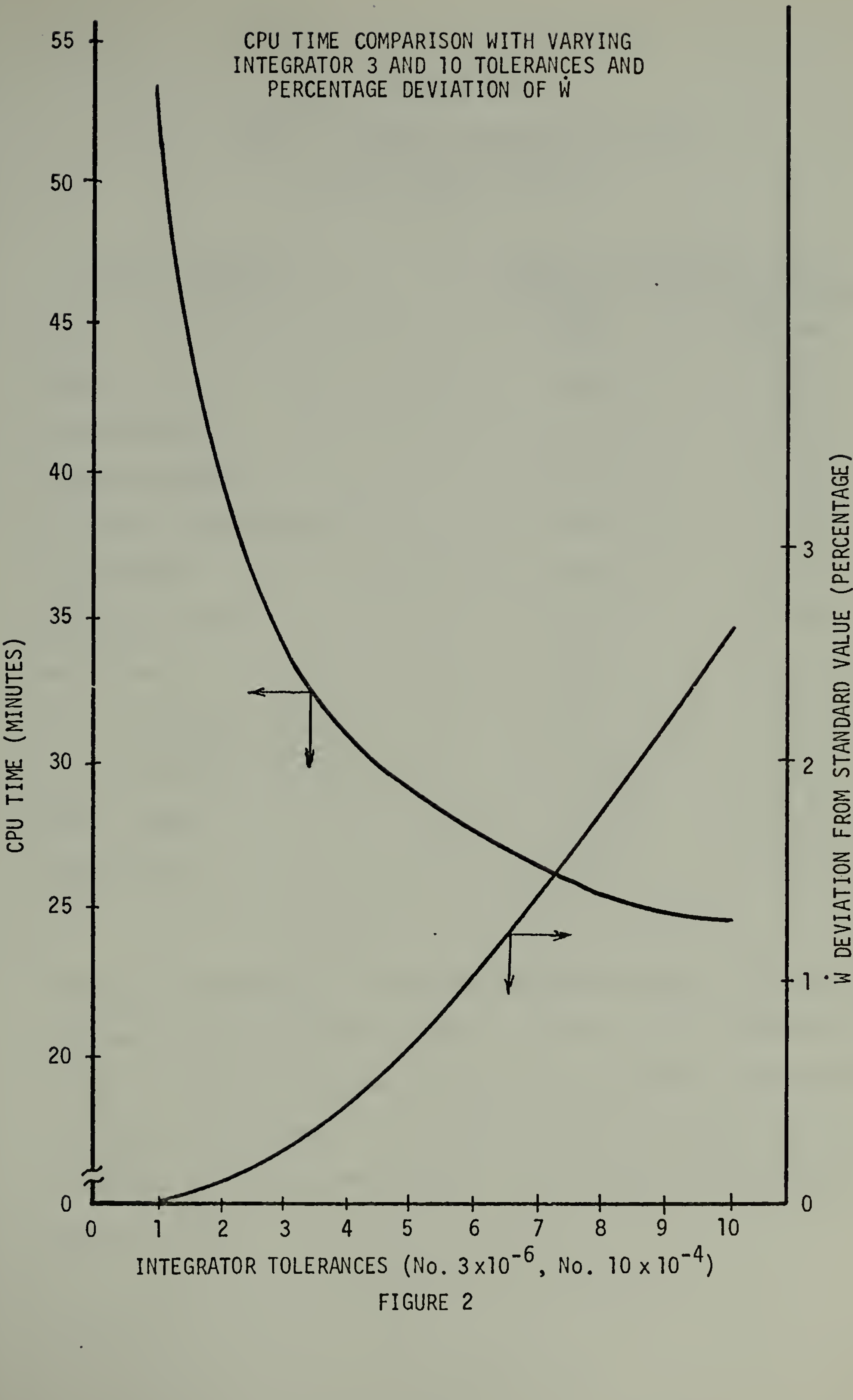

CPU TIME COMPARISON WITH VARYING
INTEGRATOR 3 AND 10 TOLERANCES AND
PERCENTAGE DEVIATION OF $\dot{W}$
CPU TIME (MINUTES)
$\dot{W}$ DEVIATION FROM STANDARD VALUE (PERCENTAGE)
0
20
25
30
35
40
45
50
55
1
2
3
0
1
2
3
4
5
6
7
8
9
10
INTEGRATOR TOLERANCES (No. 3×10^{-6}, No. 10×10^{-4})

FIGURE 2

Table IV

Output Variable Maximum Deviation
From Standard Values

Output Variable	Maximum Deviations from Standard Value (Percentage)	
	RMS	Average
Draft	0.04	0.07
Pitch Angle	0.82	0.00
Plenum Pressure	0.44	0.29
CG Heave Acceleration	2.50	0.00
Surge Speed	0.03	0.02
X-Displacement	0.04	0.04
Fan Power	1.39	3.33
Air Flow Into Plenum	0.05	0.95
Air Flow Out of Plenum	0.76	1.22
Bubble Drag	0.61	0.27
Pitch Rate	0.00	0.00

tolerance increase of a factor of seven gives a CPU time savings of 26.56 minutes, while increasing the tolerance level to a factor of ten achieves only another 2.49 minutes savings in CPU time. At the same time the percentage deviation in the value of heave acceleration increases from 1.8% to 2.5%.

2. <u>Integrators 1,2 and 4-9</u>

With integrator 3 and 10 tolerances at 0.000007 and 0.0007 respectively, each of the other integrator tolerances were increased in turn by a factor of ten. Simulated run conditions were as previously described and CPU times were compared. No appreciable CPU time savings was observed during any of these runs. The average CPU time was 27.59 minutes for these runs. The percentage deviation in the values of the output variable was negligibly small, indicating that these variables were much less susceptible to the moments and forces caused by the simulated sea state conditions.

C. DIFFERENT TYPE RUN CONDITIONS

Various run conditions were simulated in order to study the effect of loosening the tolerances on integrator 3 and 10 while maintaining standard tolerances on the other eight integrators. Each computer run simulated a 30 second craft run.

Table V shows the results of these runs with respect to CPU time savings.

Comparison of the results shown in Table III and Table V indicate that the percentage of CPU time savings is about the same regardless of the type run simulated. Therefore increasing the tolerance levels on integrators 3 and 10 by a given amount will result in a predictable amount of computer CPU time savings.

Table V

CPU Time Comparison For

Different Type Run Conditions

(Sea State 3)

Speed (knots)	Integrator Tolerance No. 3	Integrator Tolerance No. 10	Wave Encounter Angle	CPU Time (min.)	Percentage Decrease in CPU time
40	0.000001*	0.0001*	150°	52.09	Standard
	0.000007	0.0007		24.73	52.52
	0.000010	0.0010		23.47	54.94
60	0.000001*	0.0001*	Partial Turn	68.04	Standard
	0.000007	0.0007		34.34	49.24
	0.000010	0.0010		31.93	53.07
60	0.000001*	0.0001*	180° Turn	69.97	Standard
	0.000007	0.0007		36.36	48.03
	0.000010	0.0010		32.46	53.61

* Standard for comparison

In comparing results of deviations of output variables under different run conditions, no change was observed between the 40 knot, head sea simulation and the 40 knot 150° wave encounter simulation outputs. However, under the more severe conditions of 60 knots with turns applied some of the output variable values did have an increased deviation from the standard values. Table VI depicts these results for the variables concerned.

Table VI

Output Variable Maximum Deviation

From Standard Values

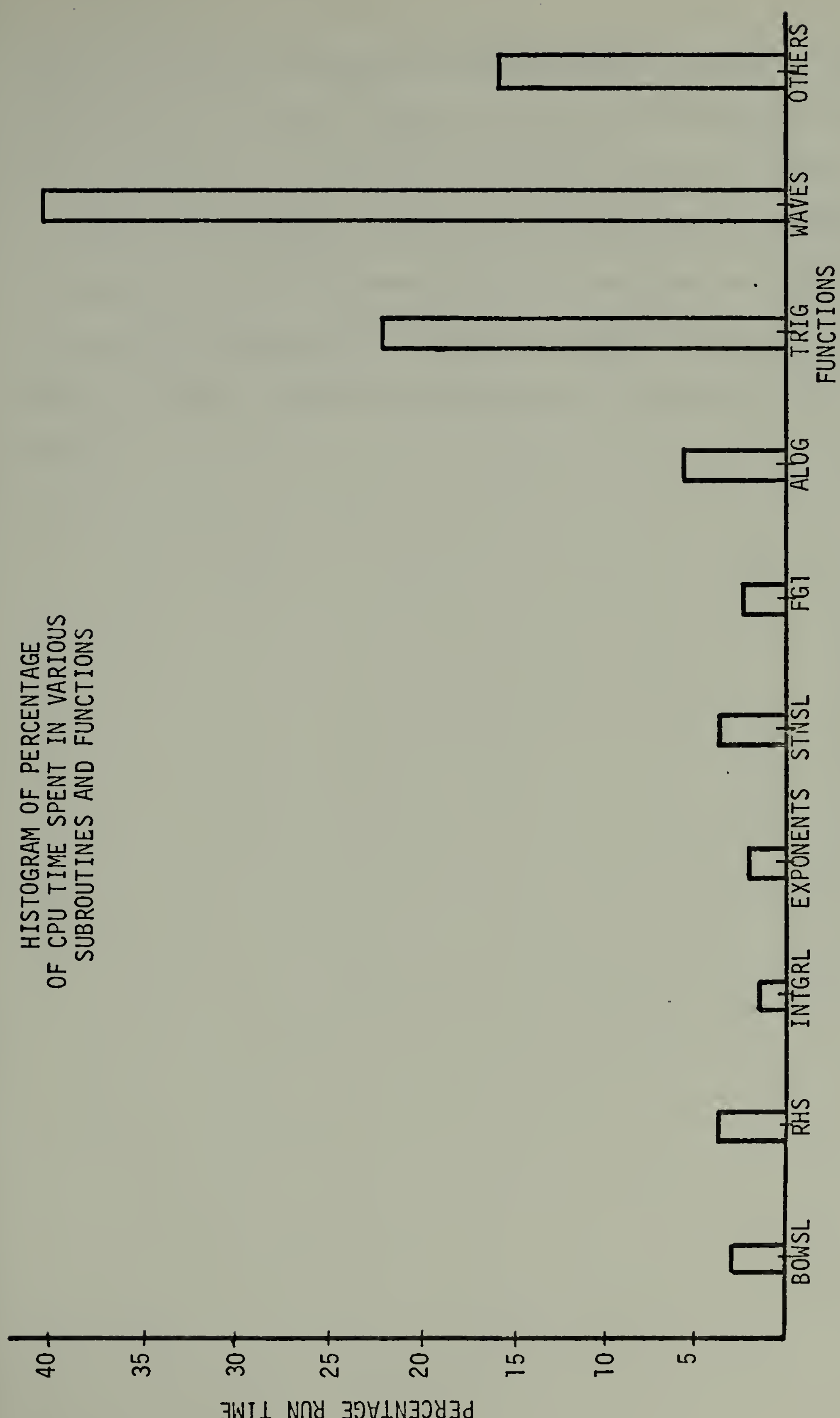
(Sea State 3, 60 knots)

Output Variable	Turn Condition	Maximum Deviation From Standard Value (Percentage)	
		RMS	Average
Plenum Pressure	Partial	1.55	0.29
	180°	1.30	0.05
CG Heave Acceleration	Partial	4.76	1.00
	180°	5.13	0.00
Fan Power	Partial	0.90	4.58
	180°	1.41	4.61
Air Flow Out of Plenum	Partial	0.17	1.69

D. SUBROUTINE PROGLOOK

In order to determine other areas of the SES simulation program where large amounts of CPU time was being used, Subroutine PROGLOOK was utilized. Integrator 3 and 10 tolerances were 0.000007 and 0.0007 respectively. A 40 knot, state 3 sea, head waves, 30 second run was simulated. Figure 3 is a histogram of the results of the run.

It was found that 41% of the computing time was spent in subroutine WAVES. The bulk of this time was consumed in the wave table. A surprisingly large amount of time



HISTOGRAM OF PERCENTAGE OF CPU TIME SPENT IN VARIOUS SUBROUTINES AND FUNCTIONS

SUBROUTINE OR FUNCTION

FIGURE 3

(22%) was spent computing trigonometric function values. The bulk of the time spent in a given subroutine was generally found to be involved in repetitious DO LOOPs.

The area labeled "Other" included various FORTRAN built in functions and the remaining subroutines and functions of the SES program which were not identified on the graph. Each of these routines contributed less than 1% of computation time.

V. CONCLUSION AND RECOMMENDATIONS

A. INTEGRATOR TOLERANCES

The computer run time dependence of the SES simulation program on the tolerance levels chosen for the Runge-Kutta-Merson algorithm is clearly demonstrated by the results shown in this thesis. Further, CPU time is directly related to the magnitude of the tolerance levels of integrators 3 and 10 and is relatively independent of the tolerance levels established for the other eight integrators.

Output variable accuracy as would be expected is also related to the tolerance levels of the integrators, however the percentage deviations from standard values is small and the values obtained could be considered adequate for most purposes. In particular, initial studies under a given set of conditions would realize a considerable amount of CPU time saving by using increased tolerance levels, while maintaining sufficiently accurate solutions to determine the relative magnitude of the forces and moments acting on the craft. As more precise values are required, the tolerance levels could be progressively tightened with an accompanying predictable increase in CPU time requirements (a valuable planning tool).

It should be noted that the individual programmer should determine his standard of accuracy for each output variable. Since the magnitude of the outputs vary greatly, a small

percentage deviation in one value may be acceptable for a given case study while being entirely too large for another output value under the same conditions. For example, the air flow rate into the plenum has a magnitude on the order of 5000 cubic feet per second with a 1% deviation from standard which may be entirely acceptable, whereas the CG heave acceleration magnitude of 0.32 "g" units with 1% deviation may not be acceptable.

Due to time limitations, the simulated runs conducted in this thesis were confined to relatively slow speeds in the 40-60 knot range. The deviations observed for the critical output variables of heave acceleration, plenum pressure, and air flow in and out of the plenum, increased with an increase in speed and with the introduction of turns. Therefore it is recommended that further studies be made at higher speeds and more severe sea state conditions to determine the validity of results obtained with increased tolerance levels.

B. DIGITAL PROGRAMMING TECHNIQUES

Time studies of the SES Loads and Motion program using subroutine PROGLOOK indicated that an appreciable amount of computer time was spent in the trigonometric functions. Most large computers use nested Chebyshev Polynomials or Taylor Series expansions as an efficient and accurate solution to the trigonometric functions [Ref. 6]. A faster method of solution, for example, would be a four place table lookup

for the functions. A careful analysis of the loss of accuracy incurred by this method would have to be conducted. Another consideration would be the increased core requirements due to the addition of a lengthy table.

It was noted that considerable computation time is spent in the execution of DO LOOPs throughout the program. It is recommended that an in depth study be made with the goal of replacing the iterative DO LOOP process with straight forward calculations where possible. An excellent example of this type of programming is presented in Appendix A of this thesis. A straight forward programming technique resulted in the elimination not only of DO LOOPs but a subroutine as well.

C. HYBRID COMPUTATIONS

Plans are being formulated at the Naval Postgraduate School with regards to the feasibility of adapting the SES Loads and motion simulation programs to a hybrid computation technique. The development of linearized equations of motion presented in Ref. 7, and the analog computer simulation developed in Ref. 8, could serve as a starting point for conversion to hybrid computation.

The Naval Postgraduate School's hybrid computation facilities include the XDS 9300 digital computer interfaced with a COMCOR 5000 analog computer and an Adage Graphics Terminal. Perhaps the most formidable problem presented by a hybrid conversion would be the core storage requirement

in the digital computer. The present storage capabilities could however be augmented by disc and/or tapes.

The parameter scaling problems referred to in Ref. 8 could be readily handled in a hybrid configuration. Other advantages to hybrid computation of the Loads and motion program include near real time solutions and the versatility of being able to change parameters during the run. The visual representation of the output parameters afforded by the Adage Graphic Terminal would be a powerful engineering tool. Hybrid computation would be particularly useful in vertical plane studies such as heave acceleration analysis and prediction, and for control systems studies.

APPENDIX A

An Example of Programming Changes to Improve Efficiency

Reference 9 presented the following changes to the SES Loads and Motions Program to improve the efficiency of the program. The affected subroutines are RHS and INCON and in addition the subroutine DMINV is deleted.

The following portion of subroutine INCON was deleted:

```
212  DO 211 I = 1,6
     DO 211 N = 1,6
211  A(I,N) = 0.0
     DO 213 N = 1,3
213  A(N,N) = AM
     A(4,4) = AIXX
     A(5,5) = AIYY
     A(6,6) = AIZZ
     A(4,6) = -AIXZ
     A(6,4) = -AIXZ
     AIMAX = AMAXI(AM,AIXX,AIYY,AIZZ,ABS(AIXZ))
     DO 214 I = 1,6
     DO 214 J = 1,6
214  A(I,J) = A(I,J)/AIMAX
     CALL DMINV(A,G,D)
     DO 215 I = 1,6
     DO 215 J = 1,6
215  A(I,J) = A(I,J)/AIMAX
     IF (D.NE.0.0)  GO TO 10
     WRITE (6,216)
     STOP
```

Inserted in place of the above programming was the following:

```
AMASSI = 1.0/AM
D = 1.0/(AIXX*AIZZ-AIXZ*AIXZ)
DIXX = AIXX*D
DIXZ = AIXZ*D
DIZZ = AIZZ*D
AIYYI = 1.0/AIYY
GO TO 10
```

The INCON parameters were linked to subroutine RHS by the following COMMON statement

```
COMMON/ATRIX/AMASSI,AIYYI,DIXX,DIXZ,DIZZ
```

In subroutine RHS the six element matrix GF(I) was deleted and the following identifiers were substituted for the summation of forces: SUMX, SUMY, SUMZ, SUMK, SUMM, SUMN. In addition the following deletion was made.

```
   DO 1 I = 1,6
   VALUE(I) = 0.0
   DO 1 J = 1,6
   VALUE(I) = VALUE(I)+A(I,J)*GF(J)
1  CONTINUE
```

Substituted for the above DO LOOPs was the following

```
VALUE(1) = SUMX*AMASSI
VALUE(2) = SUMY*AMASSI-R*U
VALUE(3) = SUMZ*AMASSI
VALUE(4) = SUMK*DIZZ+SUMN*DIXZ
VALUE(5) = SUMM*AIYYI
VALUE(6) = SUMN*DIXX+SUMK*DIXZ
```

The indicated changes deleted nine DO LOOPs and one subroutine resulting in a much more efficient program both from a time and a core requirement standpoint.

APPENDIX B

Equivalent Expressions for Yaw Acceleration

Reference 9 included the classic Euler equations of motion in six degrees of freedom and listed the assumptions under which certain terms of these equations could be disregarded as being insignificant. The question was raised as to whether the equation describing Yaw Acceleration ($\dot{R}$) in classic Euler terms was equivalent to the one defined by subroutines DMINV and RHS.

This appendix shows that, under certain assumptions, the two expresssions are equivalent.

The classic Euler equations as presented in [Ref. 9] are as follows.

$$\dot{U} = \frac{F_x}{m} - QW + RV \qquad (1)$$

$$\dot{V} = \frac{F_y}{m} - RU + PW \qquad (2)$$

$$\dot{W} = \frac{F_z}{m} - PV + QU \qquad (3)$$

$$\dot{P} = \frac{F_k\, I_{zz}}{I_{xx}I_{zz} - I_{xz}^2} + \frac{F_n\, I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} + \frac{Q\, I_{zz}}{I_{xx}I_{zz} - I_{xz}^2} \cdot \left[\frac{P(I_{xx}-I_{yy}+I_{zz})I_{xz}}{I_{zz}} - R(I_{zz} - I_{yy} + \frac{I_{xz}^2}{I_{zz}}) \right] \qquad (4)$$

$$\dot{Q} = \frac{F_m}{I_{yy}} + \frac{PR(I_{zz} - I_{xx})}{I_{yy}} + \frac{(R^2-P^2)I_{xz}}{I_{yy}} \tag{5}$$

$$\dot{R} = \frac{F_n}{I_{zz}} + \frac{PQ(I_{xx} - I_{yy})}{I_{zz}} + \frac{(\dot{P}-QR)I_{xz}}{I_{zz}} \tag{6}$$

The fundamental equations of motion for six degrees of freedom developed in Ref. 10 are as follows:

$$F_x = m[\dot{U} + QW - RV - X_G(Q^2+R^2) + Y_G(PQ-\dot{R}) + Z_G(PR+\dot{Q})] \tag{7}$$

$$F_y = m[\dot{V} + RU - PW - Y_G(R^2+P^2) + Z_G(QR-\dot{P}) + X_G(QP+\dot{R})] \tag{8}$$

$$F_z = m[\dot{W} + PV - QU - Z_G(P^2+Q^2) + X_G(RP-\dot{Q}) + Y_G(RQ+\dot{P})] \tag{9}$$

$$\begin{aligned} F_k = {} & I_{xx}\dot{P} + (I_{zz}-F_{yy})QR + m[Y_G(\dot{W}+PV-QU) - Z_G(\dot{V}+RU-PW) \\ & + X_GY_G(PR-\dot{Q}) - X_GZ_G(PQ+\dot{R}) + Y_GZ_G(R^2-Q^2)] \end{aligned} \tag{10}$$

$$\begin{aligned} F_m = {} & I_{yy}\dot{Q} + (I_{xx}-I_{zz})RP + m[Z_G(\dot{U}+QW-RV) - X_G(\dot{W}+PV-QU) \\ & + Y_GZ_G(QP-\dot{R}) - Y_GX_G(QR+\dot{P}) + X_GZ_G(P^2-R^2)] \end{aligned} \tag{11}$$

$$\begin{aligned} F_n = {} & I_{zz}\dot{R} + (I_{yy}-I_{xx})PQ + m[X_G(\dot{V}+RU-PW) - Y_G(\dot{U}+QW-RV) \\ & + Z_GX_G(RQ-\dot{P}) - Z_GY_G(RP+\dot{Q}) + Y_GX_G(Q^2-P^2)] \end{aligned} \tag{12}$$

where X_G, Y_G, and Z_G are distances from the center of gravity to the origin along the x, y, and z axis respectively.

Because of symmetry of the craft, the significant product terms of inertia are the $X_G Z_G$ terms. Under these conditions, equations (7)-(12) reduce to the following form.

$$F_x = m[\dot{U} + QW - RV] \tag{13}$$

$$F_y = m[\dot{V} + RU - PW] \tag{14}$$

$$F_z = m[\dot{W} + PV - QU] \tag{15}$$

$$F_k = I_{xx}\dot{P} + (I_{zz}-I_{yy})QR - I_{xz}(PQ+\dot{R}) \tag{16}$$

$$F_m = I_{yy}\dot{Q} + (I_{xx}-I_{zz})RP + I_{xz}(P^2-R^2) \tag{17}$$

$$F_n = I_{zz}\dot{R} + (I_{yy}-I_{xx})PQ + I_{xz}(RQ-\dot{P}) \tag{18}$$

where

$$I_{xz} = mX_G Z_G$$

Solving equations (13)-(18) for the desired derivatives yields:

$$\dot{U} = \frac{F_x}{m} - QW + RV \tag{19}$$

$$\dot{V} = \frac{F_y}{m} - RU + PW \tag{20}$$

$$\dot{W} = \frac{F_z}{m} - PV + QU \tag{21}$$

$$\dot{P} = \frac{F_k}{I_{xx}} - \left(\frac{I_{zz}-I_{yy}}{I_{xx}}\right)QR + \frac{I_{xz}}{I_{xx}}(PQ+\dot{R}) \tag{22}$$

$$\dot{Q} = \frac{F_m}{I_{yy}} - \left(\frac{I_{xx}-I_{zz}}{I_{yy}}\right)RP - \frac{I_{xz}}{I_{yy}}(P^2-R^2) \tag{23}$$

$$\dot{R} = \frac{F_n}{I_{zz}} - \left(\frac{I_{yy}-I_{xx}}{I_{zz}}\right)PQ - \frac{I_{xz}}{I_{zz}}(RQ-\dot{P}) \tag{24}$$

Equations (19)-(21) are identical to equations (1)-(3). Upon substituting equation (24) into (22) and rearranging equation (23) it can be shown that equations (22)-(24) are equivalent to equations (4)-(6).

If it is assumed that P, Q, and R are each much smaller than one, then product terms involving them may be ignored and equations (1)-(6) reduce to the following equations as stated in Ref. 9.

$$\dot{U} = \frac{F_x}{m} \tag{25}$$

$$\dot{V} = \frac{F_y}{m} - RU \tag{26}$$

$$\dot{W} = \frac{F_z}{m} \tag{27}$$

$$\dot{P} = \frac{F_k I_{zz}}{I_{xx}I_{zz} - I_{xz}^2} + \frac{F_n I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} \tag{28}$$

$$\dot{Q} = \frac{F_m}{I_{yy}} \tag{29}$$

$$\dot{R} = \frac{F_n}{I_{zz}} + \dot{P}\,\frac{I_{xz}}{I_{zz}} \tag{30}$$

The equations of motion as decoded from subroutines DMINV and RHS are as follows.

$$\dot{U} = \frac{F_x}{m} \tag{31}$$

$$\dot{V} = \frac{F_y}{m} - RU \tag{32}$$

$$\dot{W} = \frac{F_z}{m} \tag{33}$$

$$\dot{P} = \frac{F_k I_{zz}}{I_{xx} I_{zz} - I_{xz}^2} + \frac{F_n I_{xz}}{I_{xx} I_{zz} - I_{xz}^2} \tag{34}$$

$$\dot{Q} = \frac{F_m}{I_{yy}} \tag{35}$$

$$\dot{R} = \frac{F_n I_{xx}}{I_{xx} I_{zz} - I_{xz}^2} + \frac{F_k I_{xz}}{I_{xx} I_{zz} - I_{xz}^2} \tag{36}$$

Equations (25)-(29) are identical to equations (31)-(35). Substituting equation (28) into equation (30) yields:

$$\dot{R} = \frac{F_n}{I_{zz}} + \left(\frac{F_k I_{zz}}{I_{xx}I_{zz} - I_{xz}^2} + \frac{F_n I_{xz}}{I_{xx}I_{zz} - I_{xz}^2}\right)\frac{I_{xz}}{I_{zz}}$$

$$= \frac{F_n}{I_{zz}} + \frac{F_k I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} + \frac{F_n I_{xz}^2}{I_{xx}I_{zz}^2 - I_{zz}I_{xz}^2}$$

$$= \frac{F_k I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} + F_n\left(\frac{1}{I_{zz}} + \frac{I_{xz}^2}{I_{xx}I_{zz}^2 - I_{zz}I_{xz}^2}\right)$$

$$= \frac{F_k I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} + F_n\left(\frac{I_{xx}I_{zz}^2 - I_{zz}I_{xz}^2 + I_{zz}I_{xz}^2}{I_{zz}^2(I_{xx}I_{zz} - I_{xz}^2)}\right)$$

$$\dot{R} = \frac{F_k I_{xz}}{I_{xx}I_{zz} - I_{xz}^2} + \frac{F_n I_{xx}}{I_{xx}I_{zz} - I_{xz}^2} \tag{37}$$

Equation (37) is identical to equation (36) therefore, based on the given assumptions, the classical Euler equations given in Ref. 9 and as developed from Ref. 10 are mathematically equivalent to those defined in subroutines DMINV and RHS. However it is felt that the validity of disregarding product terms should be more thoroughly investigated. It is recommended that equations of the form (1)-(6) be incorporated in future digital simulation studies to determine the degree of contribution of the cross product terms on the resultant output variables.

BIBLIOGRAPHY

1. Oceanics Incorporated, Report No. 71-84A, A Study of Surface Effect Ship (SES) Craft Loads and Motions, Part I – Equations of Motion of SES Craft With Six Degrees of Freedom, Paul Kaplan, James Bentson, and Theodore P. Sargent, August, 1971.

2. Oceanics Incorporated, Report No. 71-85, SES Motions and Loads Program Users Manual, J. Bentson, T.P. Sargent, and A.I. Ruff, August, 1971.

3. Cagle, L.F., Some Performance Characteristics of the Bell 100 Ton Surface Effect Ship, M.S. Thesis, Naval Postgraduate School, June, 1973.

4. Myers, K.R., Turning Characteristics of the Bell 100 Ton Surface Effect Ship, M.S. Thesis, Naval Postgraduate School, June, 1973.

5. W.R. Church Computer Center, Naval Postgraduate School, Users Manual, 1st. ed., March, 1970.

6. IBM Systems Library, IBM System/360 Operating System FORTRAN IV (G and H) Programmers Guide, Form No. (25) GC-28-6817, 5th ed., September, 1973.

7. Oceanics Incorporated, Report No. 71-84B, A Study of Surface Effect Ship (SES) Craft Loads and Motions, Part II – Habitability and Maneuverability of SES Craft, P. Kaplan, J. Bentson, and T.P. Sargent, August, 1971.

8. Oceanics Incorporated, Report No. 73-98B, Advanced Loads and Motions Studies for Surface Effect Ship (SES) Craft, Part II – Analog Computer Simulation; M.N. Silbert and P. Kaplan, June, 1973.

9. Private Communication from J.E. Blalock to Professor A. Gerba, dated 20 April, 1973.

10. Abkowitz, M.A., Stability and Motion Control of Ocean Vehicles, The M.I.T. Press, 1972.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Assoc. Professor Alex Gerba, Jr., Code 52GZ Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	5
4.	Professor George J. Thaler, Code 52TR Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	CDR. Donald Hay, USN Surface Effect Ships Project Office, Navy Dept. P.O. Box 34401 Bethesda, Maryland 20034	1
6.	Mr. Sydney Davis, Code PMS-304-31A Department of the Navy Surface Effect Ships Project Office P.O. Box 34401 Bethesda, Maryland 20034	1
7.	Mr. James E. Blalock, Code 1607 Department of the Navy Naval Ship Research and Development Center Carderock Laboratory Bethesda, Maryland 20034	1
8.	Mr. Arnold W. Anderson, Code PMS-304-31A.1 Department of the Navy Surface Effect Ships Project Office, PM-17 Bethesda, Maryland 20034 (P.O. Box 34401)	1
9.	LCDR. William R. Mitchell CONCRUDESGRU Three FPO, San Francisco, California 96601	2

10. LCDR Michael N. Hayes, NAVSEC 6110.10 1
Naval Ship Engineering Center
Center Building, Prince Georges Center
Hyattsville, Maryland 20782

11. Mr. Reilly Comrad, Code 6136 1
Naval Ship Engineering Center
Center Building, Prince Georges Center
Hyattsville, Maryland 20782

12. Mr. Al Ford, Code 163 1
Department of the Navy
Naval Ship Research and Development Center
Carderock Laboratory
Bethesda, Maryland 20034

Thesis
M645
c.1

155314

Mitchell

Investigation of methods to optimize captured air bubble surface effect ship digital simulation for irregular sea conditions.